

RESAMPLING SYSTEM AND METHOD FOR GRAPHICS DATA INCLUDING SINE-WAVE COMPONENTS

TECHNICAL FIELD

The present invention is related generally to the field of computer
5 graphics, and more particularly, a system and method for resampling graphics data of a
source image to produce a destination image.

BACKGROUND OF THE INVENTION

As display devices of various sizes and increased resolution have been
developed and the demand for them have increased, the ability for a graphics processing
10 system to resize and resample source images and create destination images to take
advantage of the various sized and higher resolution displays is a desirable operation.
In an electronic display system, color at each pixel is represented by a set of color
components, and each color component is represented by a sample value. Color
components such as red, green, blue (RGB) or other representations such as $YCbCr$ are
15 well known in the art. Whichever representation is chosen, each color component can
be interpreted as a two dimensional array of samples, so three such arrays can represent
images on display systems. Conceptually, resampling can be viewed as a spatial
process, working on discrete input samples, represented by pixels of the source image
arranged in a two-dimensional bitmap. The output samples of the destination image are
20 spatially located at fractional sample positions within the input sample grid. Various
interpolation and modeling methods are used to construct transition models between
samples of the source image from which additional graphics data is produced during the
resampling operation.

The additional graphics data is then used to produce larger or higher
25 resolution destination graphics images. However, the resulting destination image must
retain an acceptable image quality with respect to the source image. That is, the
destination image should appear to retain at least a similar visual qualities of the source
image, such as having nearly the same color balance, contrast, and brightness as the
original source image. Otherwise, rather than accurately reproducing a larger or higher

resolution graphics image of the source image, the resampling operation will compromise image quality by introducing image distortion. To this end, various resampling algorithms have been developed in order to create high quality destination graphics images.

5 With many conventional resampling algorithms, a transition model between input samples along each axis is constructed to provide output sample values. Generally good results can be obtained with separable processing along each axis for graphics images because image feature cross-sections have the same characteristics when viewed at any angle within the image plane, only at different effective sample
10 rates. The transition models between the input samples are constructed such that the output samples interpolated from the transition model create a destination image that closely resembles the original or source image. The transition models are typically continuous so that an output sample can be generated at any position between the input samples.

15 Although an axis separable cubic model between two input samples can provide a model with very desirable reconstruction characteristics, algorithms for resampling and sharpening graphics data representing video often are not suitable for resizing and resampling graphics data representing test patterns containing sine-wave components. Such test patterns are called zone plates, and are characterized by a
20 frequency component along each axis, each of which is a function of position within the pattern. The position and frequency functions are designed to change frequencies smoothly and continuously with position.

Zone plates may be embedded within patterns testing various other attributes of a video camera, storage, transmissions or display system. They are
25 effective in testing systems with analog components (*e.g.*, analog modulated terrestrial broadcasting), and may provide some useful tests for spectrally based compression systems (such as DCTs used in MPEG). However, these tests generally do not correspond to any attributes of the human visual system. Nevertheless, the human eye is very adept at observing large areas of inconsistency in the presentation of these
30 patterns. Thus, to avoid viewer complaints or feelings of disappointment (whether or

not they are justified), a graphics processing system having resampling and resizing capabilities should be able to accommodate these test patterns.

Therefore, there is a need for a method and system for resampling graphics data of images having sine-wave components.

5 SUMMARY OF THE INVENTION

The present invention relates to a method and system for calculating resample output values from input samples and their associated sample values. A resampling circuit calculates a frequency value for a sine-wave model from a sample set of the input samples and determines whether the frequency value is in a frequency
10 range. In the case where the frequency value is in the frequency range, a sinusoidal transition model is determined based on the sample set. However, if the frequency value is outside of the frequency range, a non-sinusoidal model is determined based on the sample set. The resampling circuit then calculates resample output values from the resulting sinusoidal or non-sinusoidal model.

15 BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system in which embodiments of the present invention are implemented.

Figure 2 is a block diagram of a graphics processing system in the computer system of Figure 1.

20 Figure 3 is a block diagram of a resampling circuit in the graphics processing system of Figure 2 according to an embodiment of the present invention.

Figure 4 is a diagram representing a sample of graphics data and corresponding sample values.

DETAILED DESCRIPTION OF THE INVENTION

25 Embodiments of the present invention provide a method and system for calculating resampled values from a source graphics image having graphics data including sine-wave components. Certain details are set forth below to provide a

sufficient understanding of the invention. However, it will be clear to one skilled in the art that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, timing protocols, and software operations have not been shown in detail in order to avoid unnecessarily obscuring the invention.

Figure 1 illustrates a computer system 100 in which embodiments of the present invention are implemented. The computer system 100 includes a processor 104 coupled to a host memory 108 through a memory/bus interface 112. The memory/bus interface 112 is coupled to an expansion bus 116, such as an industry standard architecture (ISA) bus or a peripheral component interconnect (PCI) bus. The computer system 100 also includes one or more input devices 120, such as a keypad or a mouse, coupled to the processor 104 through the expansion bus 116 and the memory/bus interface 112. The input devices 120 allow an operator or an electronic device to input data to the computer system 100. One or more output devices 120 are coupled to the processor 104 to provide output data generated by the processor 104. The output devices 124 are coupled to the processor 104 through the expansion bus 116 and memory/bus interface 112. Examples of output devices 124 include printers and a sound card driving audio speakers. One or more data storage devices 128 are coupled to the processor 104 through the memory/bus interface 112 and the expansion bus 116 to store data in, or retrieve data from, storage media (not shown). Examples of storage devices 128 and storage media include fixed disk drives, floppy disk drives, tape cassettes and compact-disc read-only memory drives.

The computer system 100 further includes a graphics processing system 132 coupled to the processor 104 through the expansion bus 116 and memory/bus interface 112. Optionally, the graphics processing system 132 may be coupled to the processor 104 and the host memory 108 through other types of architectures. For example, the graphics processing system 132 may be coupled through the memory/bus interface 112 and a high speed bus 136, such as an accelerated graphics port (AGP), to provide the graphics processing system 132 with direct memory access (DMA) to the host memory 108. That is, the high speed bus 136 and memory bus interface 112 allow

the graphics processing system 132 to read and write host memory 108 without the intervention of the processor 104. Thus, data may be transferred to, and from, the host memory 108 at transfer rates much greater than over the expansion bus 116. A display 140 is coupled to the graphics processing system 132 to display graphics images. The display 140 may be any type of display, such as a cathode ray tube (CRT), a field emission display (FED), a liquid crystal display (LCD), or the like, which are commonly used for desktop computers, portable computers, and workstation or server applications.

Figure 2 illustrates circuitry included within the graphics processing system 132 for performing various three-dimensional (3D) graphics functions. As shown in Figure 2, a bus interface 200 couples the graphics processing system 132 to the expansion bus 116. In the case where the graphics processing system 132 is coupled to the processor 104 and the host memory 108 through the high speed data bus 136 and the memory/bus interface 112, the bus interface 200 will include a DMA controller (not shown) to coordinate transfer of data to and from the host memory 108 and the processor 104. A graphics processor 204 is coupled to the bus interface 200 and is designed to perform various graphics and video processing functions, such as, but not limited to, generating vertex data and performing vertex transformations for polygon graphics primitives that are used to model 3D objects. The graphics processor 204 is coupled to a triangle engine 208 that includes circuitry for performing various graphics functions, such as clipping, attribute transformations, rendering of graphics primitives, and generating texture coordinates for a texture map.

A pixel engine 212 is coupled to receive the graphics data generated by the triangle engine 208. The pixel engine 212 contains circuitry for performing various graphics functions, such as, but not limited to, texture application or mapping, bilinear filtering, fog, blending, and color space conversion. A memory controller 216 coupled to the pixel engine 212 and the graphics processor 204 handles memory requests to and from an local memory 220. The local memory 220 stores graphics data, such as source pixel color values and destination pixel color values. A display controller 224 is coupled to the memory controller 216 to receive processed destination color values for pixels that are to be rendered. Coupled to the display controller 224 is a resampling

circuit 228 that facilitates resizing or resampling graphics images. As will be explained below, embodiments of the resampling circuit 228 perform approximations that simplify the calculation of a model between two sample points for use during resampling. The output color values from the resampling circuit 228 are subsequently
 5 provided to a display driver 232 that includes circuitry to provide digital color signals, or convert digital color signals to red, green, and blue analog color signals, to drive the display 140 (Figure 1).

Although the resampling circuit 228 is illustrated as being a separate circuit, it will be appreciated that the resampling circuit 228 may also be included in one
 10 of the aforementioned circuit blocks of the graphics processing system 132. For example, the resampling circuit 228 may be included in the graphics processor 204 or the display controller 224. In other embodiments, the resampling circuit 228 may be included in the display 140 (Figure 1). Therefore, the particular location of the resampling circuit 228 is a detail that may be modified without deviating from the
 15 subject matter of the invention, and should not be used in limiting the scope of the present invention.

Figure 3 illustrates a resampling circuit 300 that may be substituted for the resampling circuit 228 shown in Figure 2. The resampling circuit 300 includes a sine-model resampling circuit 312 for determining if a sample of graphics data provided
 20 by the display driver 224 (Figure 2) is likely to include sine-wave components. As mentioned previously, the resampling operations used for resampling graphics data including sine-wave components is often different than that used for resampling other graphics data representing other types of graphics images, such as video. Although one resampling algorithm may be used, the image quality of one or the other types of
 25 graphics data will be compromised. Thus, two different resampling operations are used for the different types of graphics data, one for graphics data including sine-wave components performed by the sine-model resampling circuit 312 and one for non-sine-wave graphics data performed by a non-sine-wave resampling circuit 308.

It will be appreciated that the sample values for the samples may consist
 30 of several different components. For example, the sample value may represent pixel

colors which are the combination of red, green, and blue color components. Another example includes sample values representing pixel colors which are the combination of luma and chroma components. Consequently, because it is well understood in the art, although circuitry to perform graphics operation for each of the components is not

5 expressly shown or described herein, embodiments of the present invention include circuitry, control signals, and the like necessary to perform resampling operations on each component for multi-component sample values. Moreover, it will be appreciated that embodiments of the present invention further include the circuitry, control signals, and the like necessary to perform axis separable resampling operations for graphics data

10 represented in multiple axes. Implementation of axis separable resampling is well understood in the art, and a more detailed description of such has been omitted from herein to avoid unnecessarily obscuring the present invention.

The non-sine-wave resampling circuit 308 can perform conventional resampling operations that are well known to those of ordinary skill in the art.

15 Alternatively, a resampling operation such as that described in co-pending application having U.S. Serial Number _____, entitled PIXEL RESAMPLING SYSTEM AND METHOD to Slavin, which is incorporated herein by reference, can also be performed by the non-sine-wave resampling circuit 308. In summary, the subject matter of the aforementioned patent application includes generating a cubic

20 model for transitions between adjacent samples from the sample values and the gradient values cosited with the two samples. The cosited gradients are approximated to facilitate generation of the transition model. The coefficients for the cubic model are determined from the known values and used by a cubic model evaluation circuit to calculate resampled values between the adjacent samples. As will be explained in more

25 detail below, the cubic model evaluation circuit described in the aforementioned patent application may be used with the present invention to determine resampled values for graphics data including sine-wave components.

In operation, when a resampling operation is to be performed, the resampling circuit 228 (Figure 2) becomes active and sine-model resampling circuit 312

30 receives sample values for the samples of graphics data of the source image to be

resampled. As will be explained in more detail below, based on a sampling of the graphics data received by the resampling circuit 300, the sine-model resampling circuit 312 determines whether the graphics data includes sine-model components. If so, the sine-model resampling circuit 312 performs the resampling operation on the graphics data. All other graphics data is provided to the non-sine-model resampling circuit 308 for the resampling operation. The sine-model resampling circuit 308 performs operations of fitting a sine-model to the sample of graphics data it receives. Each of the respective resampling circuits perform various operations to resample the graphics data received from display controller 224 (Figure 2) to produce a resampled graphics image.

The resampled data generated by the resampling circuits are subsequently used in the process of rendering enlarged or higher resolution graphics images.

Although graphics data including sine-wave components may change frequency with position, such as in a zone plate test pattern, the sine-model resampling circuit 312 performs the operation with localized processing. Thus, the zone plate can be regarded as having a fixed frequency in each axis over a small region. For the small region, algorithms can be used to find the parameters for the equation:

$$V_p = A \sin(\omega p + \phi) + B$$

where p is a local input sample position value along each axis, and V_p is an input sample value at position p . Although the previous equation has four unknowns, and consequently requires only four adjacent sample values, for reasons that will be explained later, we use five samples along each axis with a position index p of zero as the center of the samples. Initially, a set of four samples $S_{0...3}=V_{-2...1}$ is selected. The values of the selected sample set are used to solve the following equations to obtain angular frequency ω :

25

$$\begin{aligned} d_1 &= S_0 - S_3 \\ d_2 &= S_1 - S_2 \\ \cos w &= \frac{\left(\frac{d_1}{d_2} - 1\right)}{2} \\ \text{if } (\cos w) < -0.95 &\quad \text{then } \omega = \arccos(-0.95), \\ \text{else if } (\cos w) < 0.9 &\quad \text{then } \omega = \arccos(\cos w) \quad \text{else } \text{NOT-A-SINE}. \end{aligned}$$

The value of the angular frequency ω is limited to $\omega \geq \arccos(-0.95)$ to prevent the value from going too near $\pi = \arccos(-1)$, the maximum angular frequency which causes ill-conditioned behavior at later stages of processing. Although the frequency limit $\omega \geq \arccos(-0.95)$ may introduce minor errors during the following sine-model fit operation,
 5 which will be described below, the frequency limit creates the appearance of a gradual and benign “fade-out” on zone-plate patterns near π . It will be appreciated, however, that limit values nearer to -1 are possible with low-noise, higher accuracy data.

In the case where d_2 is zero, the samples are positioned symmetrically around a peak midway between samples S_1 and S_2 . Such a situation presents an infinity
 10 of sine-wave solutions, and consequently, poorly conditioned equations. However, as shown in Figure 4, this issue may be addressed by providing the sine-model resampling circuit 312 (Figure 3) using five samples instead of four. That is, d_2 is evaluated by the sine-model resampling circuit 312 from the middle two samples for each of the candidate sets of four samples:

$$\begin{aligned} 15 \quad S_{0...3} &= V_{-2...1} \\ S_{0...3} &= V_{-1...2} \end{aligned}$$

The set $\{V\}$ with the largest $|d_2|$ is selected and used to obtain a reliable estimate of the angular frequency ω .

If the maximum of d_2 from both sets of four samples still results in an
 20 angular frequency ω that is near 0 (*i.e.*, $\cos(\omega) > 0.9$), then the samples are ill-conditioned, most likely the result from sine-waves components of very low amplitude or frequency. As a result, a NOT-A-SINE error is returned for the five samples. It will be appreciated that the limit of $\cos(\omega) > 0.9$ may be modified for different noise and accuracy conditions. However, using the present limit will typically result in one of the
 25 sets of sample values $\{S\}$ yielding a useful d_2 value, and thus, provide good measurement results. Where a NOT-A-SINE error is produced, the graphics data is provided to the non-sine-model resampling circuit 308 where an alternative interpolation algorithm is performed instead. As mentioned previously, various suitable interpolation algorithms may be performed there.

Once a set of $\{S\}$ values has been selected by the sine-model resampling circuit 312 and the angular frequency ω obtained, a sine fit can be obtained by finding $\{A, \phi, B\}$ from the sine-model equation:

$$V_p = A \sin(\omega p + \phi) + B.$$

5 The values for amplitude A , phase ϕ , and offset B can be solved by the sine-model resampling circuit 312 using values that are already known, namely, the angular frequency ω , and the sample values of the middle three samples $\{V_{-1}, V_0, V_1\}$ of the five samples previously mentioned. While it would be possible to perform a least-squares fit to more than three samples, using a three-sample fit provides the benefit of
10 simplicity, and additionally, ensures that the resulting model will go through the original three sample points. Moreover, as will be explained in further detail below, additional tests can be performed by the sine-model resampling circuit 312 on the resulting three sample fit model to confirm that it is not fitting sine-models to transitions between samples of graphics data not including sine-wave components. Solving the three-
15 sample point equations results in:

$$\begin{aligned} A \cos(\phi) &= ACOS = \frac{V_1 - V_{-1}}{2 \sin(\omega)} \\ A \sin(\phi) &= ASIN = \frac{V_1 + V_{-1} - 2V_0}{2(\cos(\omega) - 1)} \\ B &= V_0 - ASIN \end{aligned}$$

which provides the offset B directly. The phase and amplitude can then be obtained
20 directly through a rectangular to polar coordinate conversion:

$$\begin{aligned} \phi &= \arctan 2(ASIN, ACOS) \\ A &= \sqrt{(ASIN)^2 + (ACOS)^2}. \end{aligned}$$

After the sine-model resampling circuit 312 resolves the $\{A, \phi, B\}$ values from the previous equations, the resulting sine-model can be evaluated to directly obtain
25 resampled values from the source image. Note that the phase ϕ is coincident with the middle of the three samples values V_0 , and that the four-quadrant $atan2(y,x)$ function is used. Further note that in the case where $\omega=0$ or $\omega=\pi$, a division by zero occurs. However, these values should have been excluded previously.

An alternative approach to determining resampled values according to a sine-model results from applying the ASIN and ACOS values used in resolving the offset value B . Expanding the sine-model equation discussed earlier results in:

$$R_p = A \sin(\phi) \cos(\omega p) + A \cos(\phi) \sin(\omega p) + B$$

5 where $R_p = V_p$ for $p = \{-1, 0, 1\}$. As discussed previously, the values for $A \sin(\phi)$ and $A \cos(\phi)$, and the angular frequency ω were determined to calculate the offset value B . Thus, R_p can be evaluated at any fractional position $p = \Delta p$ by substituting these values into the expanded sine-model equation to obtain a resampled result in each axis between the samples V_{-1} and V_0 .

10 As a means of verifying the accuracy of the sine-model generated through the three samples $\{V_{-1}, V_0, V_1\}$, the model is evaluated at the positions of the first and last of the five samples (*i.e.*, at positions V_{-2} and V_2) using the following equation:

$$\begin{aligned} diff_A &= |R_{-2} - V_{-2}| \\ diff_B &= |R_2 - V_2| \\ threshold &= \frac{A}{4} \\ \text{if } ((diff_A > threshold) \text{ or } (diff_B > threshold)) \text{ then } & \text{NOT-A-SINE.} \end{aligned}$$

15

The threshold value is set to a fraction of the amplitude of the fitted sine wave, which allows for some noise and distortions due to assumptions that the angular frequency ω is constant, or that ω may have been limited near π as previously discussed. A scaling value of $\frac{1}{4}$ works quite well, and is easy to implement, but it will be appreciated that other values are possible depending upon noise levels. This test rejects fits on edges because the outlying samples will fit badly to a sine-model which was fitted to the central three samples $\{V_{-1}, V_0, V_1\}$.

20

25 Note that $\cos(-x) = \cos(x)$ and $\sin(-x) = -\sin(x)$. Consequently, $\cos(-2\omega)$ and $\sin(-2\omega)$ can be calculated by sharing look-up tables when obtaining R_{-2} and R_2 . Moreover, $diff_A$ can be determined using two ROM tables to obtain the $\sin(2\omega)$ and $\cos(2\omega)$ values, along with two multiplies and two adders. As just discussed, only two more multipliers and adders are needed to obtain $diff_B$.

As an alternative, rather than calculating the amplitude A precisely using the equation:

$$A = \sqrt{(ASIN)^2 + (ACOS)^2}$$

which involves division operations and makes the calculations more difficult and complex to solve, a usable amplitude A can be approximated for the verification operation because the value is used only as a threshold for determining the accuracy of the resulting sine-model. An economical approximation of the amplitude A to better than 5% accuracy can be obtained using:

$$\begin{aligned} s &= |ASIN| \\ c &= |ACOS| \\ \text{if}(s > c) \quad A &\approx s + \frac{c}{2} \quad \text{else} \quad A \approx c + \frac{s}{2}. \end{aligned}$$

An incrementer (for 2's complement negation), and a multiplexer can be used to obtain the absolute value of s and c . A compare, a multiplexer, and an adder are used for the remaining operations.

As mentioned previously, resampled values for a sine-model may be directly determined from the sine-model equation:

$$R_p = A \sin(\phi) \cos(\omega p) + A \cos(\phi) \sin(\omega p) + B.$$

However, the arithmetic for directly obtaining the resampled value is relatively complex, so the resulting system is expensive in hardware. As an alternative to solving the sine-model directly, a cubic model system may be used to determine resampled values. This method of determining the resampled values may be desirable where a resampling circuit is equipped with an cubic model evaluation block. The resampling operation employs a conventional cubic evaluation circuit, which is well known in the art. Although not described in greater detail herein, implementation of a cubic model evaluation block is well understood by those of ordinary skill in the art, and the description provided herein is sufficient to allow one to practice the invention without undue experimentation. Additionally, as mentioned previously, a cubic evaluation circuit suitable for implementing embodiments of the present invention is included in the system described in the aforementioned co-pending patent application,

PIXEL RESAMPLING SYSTEM AND METHOD.

A cubic model may be used between two input samples p and $p+1$ to provide a continuous model having desirable reconstruction characteristics for graphics images. A piece-wise cubic polynomial model along an axis will be valid over a fractional input sample position Δp from 0 to 1. Consequently, the model is valid from
 5 integer sample position p to $p+1$:

$$f(P + \Delta p) = \sum_{i=0}^3 C[P, i](\Delta p)^i .$$

The resulting cubic model will go through the two input samples p and $p+1$.

As is well known, a cubic model can be solved with four constraints. Two of these constraints may be provided by the sample values f_p and f_{p+1} at the two
 10 input samples p and $p+1$. These sample values are known. Two additional constraints may be provided by the gradients gr_p and gr_{p+1} at, or co-sited with, the two input samples p and $p+1$. To solve the cosited gradients, the equation for the cubic model is differentiated with respect to Δp , resulting in:

$$gr(P + \Delta p) = \sum_{i=0}^3 iC[P, i](\Delta p)^{i-1} .$$

15 Evaluating the two equations at $\Delta p = \{0, 1\}$, and solving for the four coefficients $C[P, i]$ at the relative positions of the contributors to the cubic model are of interest results in coefficients:

$$\begin{aligned} k &= f_1 - f_0 \\ C_3 &= gr_1 + gr_0 - 2k \\ 20 \quad C_2 &= k - C_3 - gr_0 \\ C_1 &= gr_0 \\ C_0 &= f_0 \end{aligned}$$

for the cubic equation:

$$f(\Delta p) = \sum_{i=0}^3 C_i(\Delta p)^i .$$

25 The resulting cubic equation, along with the gradients gr_0 and gr_1 and the sample values f_0 and f_1 for the two input samples p and $p+1$ provides a piece-wise continuous model for resampling.

Differentiating the sine-model equation with respect to the angular frequency ω to find the gradients gr_p results in:

$$gr_p = -A \sin(\phi) \times \omega \sin(\omega p) + A \cos(\phi) \times \omega \cos(\omega p).$$

This model can obtain valid gradients at position $p = \{-1, 0, 1\}$, cosited with the original fitted samples. The gradients are then passed to the cubic evaluation block to generate a resampled output point. This approach is less accurate than calculating resampled values directly through a sine-model fit because the cubic interpolation system cannot approximate the significant higher order polynomial terms in Δp that are present in sine waves at higher frequencies. This distortion along the x-axis further compounds errors along the y-axis. However, good results can be obtained up to near 0.9 of the Nyquist sampling limit. Moreover, although two output values (gradients) are evaluated instead of one for the sine model case, the values are cosited with the input samples at discrete sample times, so as p is an integer, the hardware to evaluate gr_p is much simpler. Note that the cubic evaluation circuit which follows should be there in any case for non-sinusoids.

As mentioned previously, embodiments of the invention have been described herein with sufficient detail to allow a person of ordinary skill in the art to practice the invention. Implementation of many of the algorithms previously described may be implemented by conventional circuitry. For example, determining the angular frequency ω can be implemented using logarithm ROMs, and the corresponding anti-logarithm and limit detection can be built into another ROM. Thus, only three ROMs and three address to obtain ω once a data $\{S\}$ set has been selected. Another example is using a comparator to determine the largest $|d2|$ calculated for the two sets of samples and a multiplexer to select the final data set of $\{S\}$ to estimate the angular frequency ω . Thus, in order to prevent unnecessarily obscuring the invention, a more detailed description of the implementation of various aspects of the invention have been omitted from herein.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration,

various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.

1. A method of determining a value of a function of a variable, the method comprising:
receiving a value of the variable;
determining a value of the function of the variable based on the received value;
outputting the determined value of the function of the variable.